<u>Amendments to the Claims:</u>

This listing of claims will replace all prior versions, and

listings, of claims in the application:

<u>Listing of Claims:</u>

Claims 1 - 12 (canceled)

Claim 13 (currently amended): A method of configuring a

language of a computer program, the method which comprises the

following steps:

    selecting a text memory wherein alphanumeric message

character strings are associated with alphanumeric

identification expressions;

    <u>checking each dialog element of a user interface of the</u>

<u>computer program to determine whether a character string</u>

<u>present in the respective dialog element includes a wildcard</u>

<u>character string; and</u>

    finding in the text memory identification expressions

associated with <u>the</u> wildcard character <u>string</u> ~~strings~~

contained in the computer program and replacing the wildcard

character <u>string</u> ~~strings~~ in the computer program with the

Applic. No. 10/574,064
Response Dated June 17, 2008
Responsive to Office Action of April 1, 2008

associated message character strings in the text memory, and

thereby:

carrying out the finding and replacing steps during

a runtime of an executable binary computer program; and

carrying out the replacing step by associating the

message character strings with memory variables in the

running computer programs.

Claim 14 (previously presented): The method according to

claim 13, which comprises selecting the text memory such that

the identification expressions contain alphanumeric name

descriptors and alphanumeric field descriptors, and a

respective field descriptor has an associated message

character string.

Claim 15 (previously presented): The method according to

claim 14, wherein an identification expression in the text

memory is found for a wildcard character string contained in a

computer program by evaluating a path for the wildcard

character string, and wherein the path is formed from at least

one of the name descriptors.

Applic. No. 10/574,064
Response Dated June 17, 2008
Responsive to Office Action of April 1, 2008

Claim 16 (previously presented): The method according to claim 13, which comprising selecting the XML format for configuring the text memory, and finding the identification expressions by interpreting XML tags.

Claim 17 (previously presented): The method according to claim 16, which comprises storing identification expressions and message texts in an XML table in the XML text memory.

Claim 18 (previously presented): The method according to claim 13, which comprises reading the respective wildcard expressions to be replaced from a memory variable in a dialog structure in the computer program.

Claim 19 (currently amended): A method of configuring a language of a computer program, the method which comprises the following steps:

selecting a text memory wherein alphanumeric message character strings are associated with alphanumeric identification expressions;

checking each dialog element of a user interface of the computer program to determine whether a character string

Page 4 of 18

<u>present in the respective dialog element includes a wildcard</u>

<u>character string; and</u>

running the computer program by runtime execution of an

executable binary program file and, during the execution:

finding in the text memory identification

expressions associated with <u>the wildcard</u> character <u>string</u>

~~strings~~ contained in the computer program and replacing

the wildcard character <u>string</u> ~~strings~~ in the computer

program with the associated message character strings in

the text memory, by assigning the message character

strings to memory variables in the running computer

programs.

Claim 20 (currently amended): A computer system with means

for configuring the language of a computer program stored in

the computer system, comprising:

a text memory having stored therein an association

between alphanumeric identification expressions and

alphanumeric message character strings;

means for checking each dialog element of a user

interface of the computer program to determine whether a

character string present in the respective dialog element

includes a wildcard character string; and

means for finding identification expressions in said text

memory associated with said wildcard character string ~~strings~~

contained in the computer program and for replacing ~~the~~ said

wildcard character string ~~strings~~ in the computer program with

the associated message character strings in said text memory;

and

~~wherein~~ the computer program ~~is~~being in executable binary

code and said means for finding and replacing ~~are~~being

contained in the computer program.

Claim 21 (previously presented): The computer system

according to claim 20, wherein the identification expressions

contained in said text memory contain at least one

alphanumeric name descriptor and at least one alphanumeric

field descriptor, and a respective field descriptor has an

associated message character string.

Claim 22 (currently amended): The computer system according to claim 21, wherein the wildcard character string ~~strings~~ contained in the computer program ~~have~~ has a respective path formed from at least one of said name descriptors.

Claim 23 (previously presented): The computer system according to claim 20, wherein said text memory is in XML format, and wherein name descriptors are shown as XML tag names and field descriptors are shown as XML attribute names.

Claim 24 (previously presented): The computer system according to claim 20, wherein a respective wildcard character string contains at least one XML tag name, and the wildcard character string starts with a characteristic prefix.

Claim 25 (previously presented): The computer system according to claim 20, wherein the wildcard character strings to be replaced are stored in a memory variable in a dialog structure in the computer program.